Load Balancing for Distributed Battlefield Simulations: Initial Results

David R. Pratt, Ph.D.
SAIC
pr[illegible]ic.com

Amy E. Henninger, Ph.D.
Soar Technology, Inc.
amy@soartech.com

SAIC
An Employee-Owned Company

Soar Technology
Thinking...*inside* the box

Distribution Statement A: Approved for public release; distribution is unlimited

Why Load Balance?

Optimize resource utilization

Distribution Statement A: Approved for public release; distribution is unlimited

Load Balancing Research

Source	Vaughan and O'Donovan (1998)	Wilson and Shen (1998)	Willebeek-Lamair and Reeves (1993)	Pratt and Henninger (2003)
Load Balancing Algorithm Components	**Information policy** **Transfer policy**	**Level 1 (pending load, cpu utilization, processor load)**	**Processor Load Evaluation** **Load Balancing Profitability Determination**	**Determine load (current and projected)**
	Selection policy **Location policy**	**Level 2 (selecting load, packing load, initiating migration, migrating load,**	**Task Migration Strategy**	**Select load to migrate**
	Acceptance policy	**unpacking load, and updating global information)**	**Task Selection Strategy**	**Migrate the load**

Distribution Statement A: Approved for public release; distribution is unlimited

Load Balancing Algorithm

- Update local load measure
- Update remote load measure
- If own_load > destination_load then
- Increase Load
- Else
- Decrease Load
- Schedule next load balancing event
- Continue

Distribution Statement A: Approved for public release; distribution is unlimited

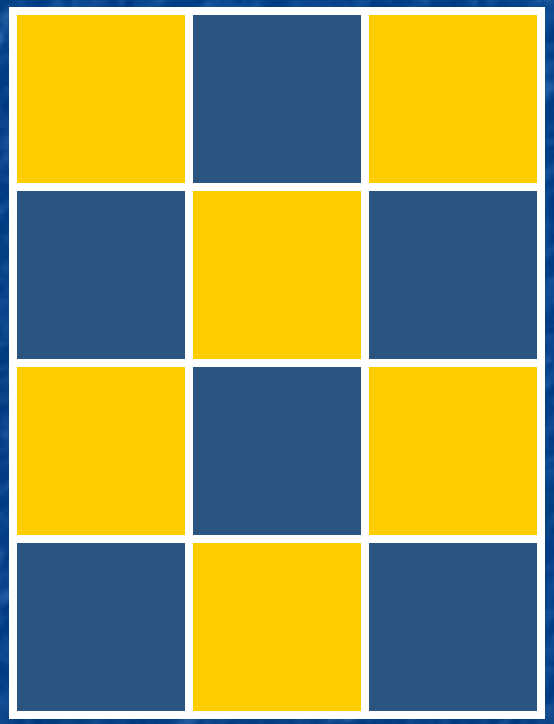
Migration Policy

- Random – will move an object if it feels like it
- MigPolicyGeographic – will move object if it resides in AOI that is governed by remote node
- MigPolicyLoad – will move object if another node has a lighter load
- MigPolicyProximity – will move an object if it is “nearest neighbor” to another node’s center of mass

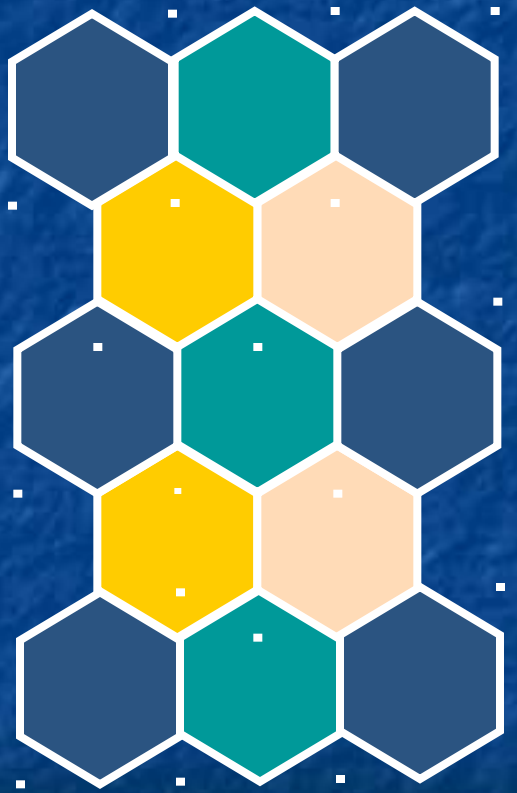
Distribution Statement A: Approved for public release; distribution is unlimited

Approaches used in Partitioning Spatial Problems

One D



Two D



Hex

Distribution Statement A: Approved for public release; distribution is unlimited

Migration Selection

- Random – randomly selects where the object will be moved.
- MigSelectionFixedAOI – will move object to node that governs AOI object resides in
- MigSelectionLightest – will move object to node with lightest load
- MigSelectionClosest – will move object to node that hosts object’s nearest neighbor center of mass

Distribution Statement A: Approved for public release; distribution is unlimited

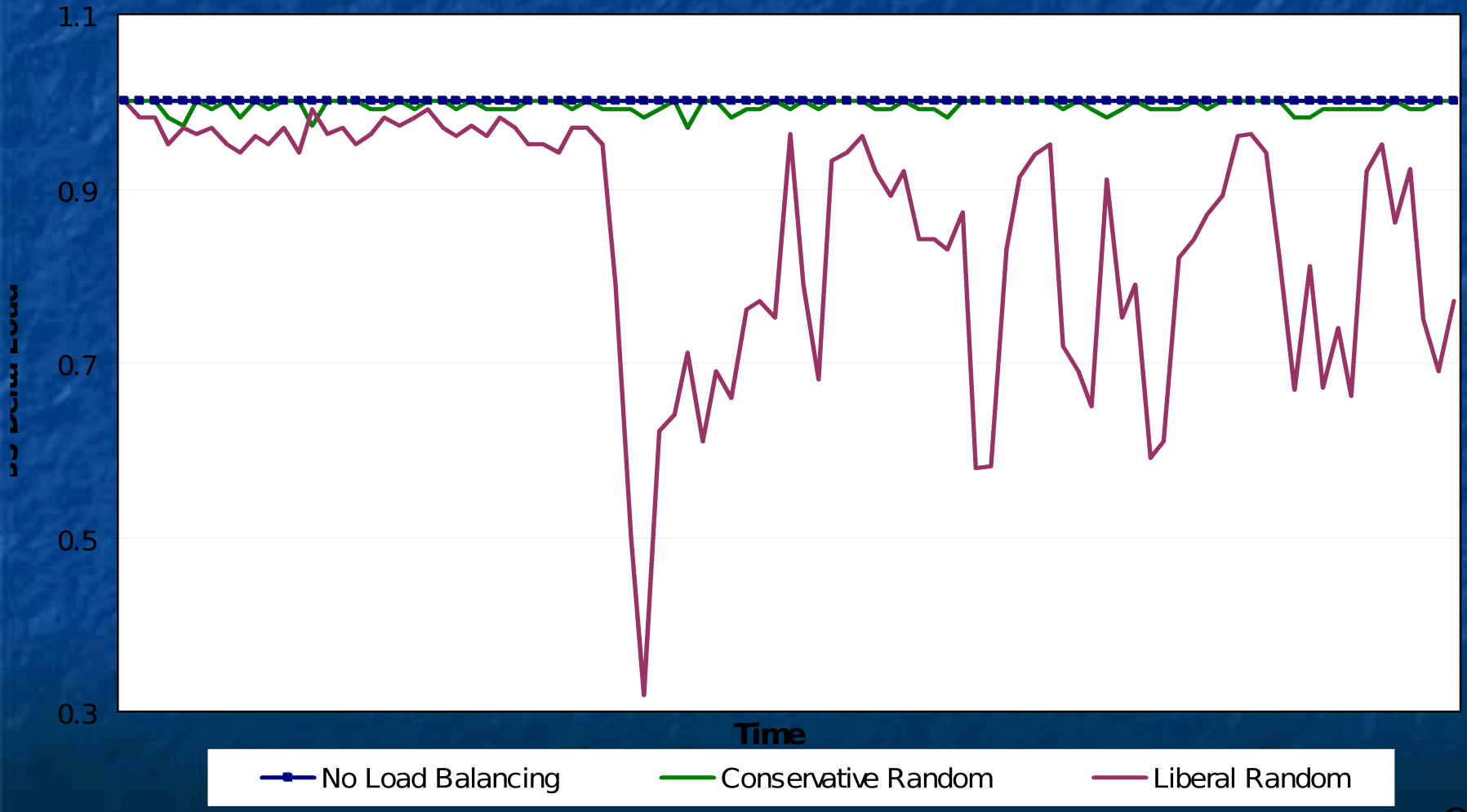
Scheduler Implementation

```
protected void processEvents(long stopTime){
  long curTime;   //the current time
  boolean idleFlag = false;             //are we processing events
  long idleStart = clock.getTime(); //start of idle period
  while ((curTime = clock.getTime()) < stopTime){    //run for the specified time
    if ( (sched.theQ.size() > 0) &&
        (sched.theQ.getNextTime() <= curTime)){       //check for event
   if(idleFlag == true){     //has the system been idle
   idleFlag = false;           //reset flag
       busytimes.add(new TimeRec(idleStart,curTime));          //record the idle time
     }
     sched.theQ.dequeue().run(curTime);     //process the events
   } else {
   if (idleFlag == false){   //start recording idle time
       idleStart = curTime;
       idleFlag = true;
     }}
  }}
```

Modified Discrete Event Scheduler

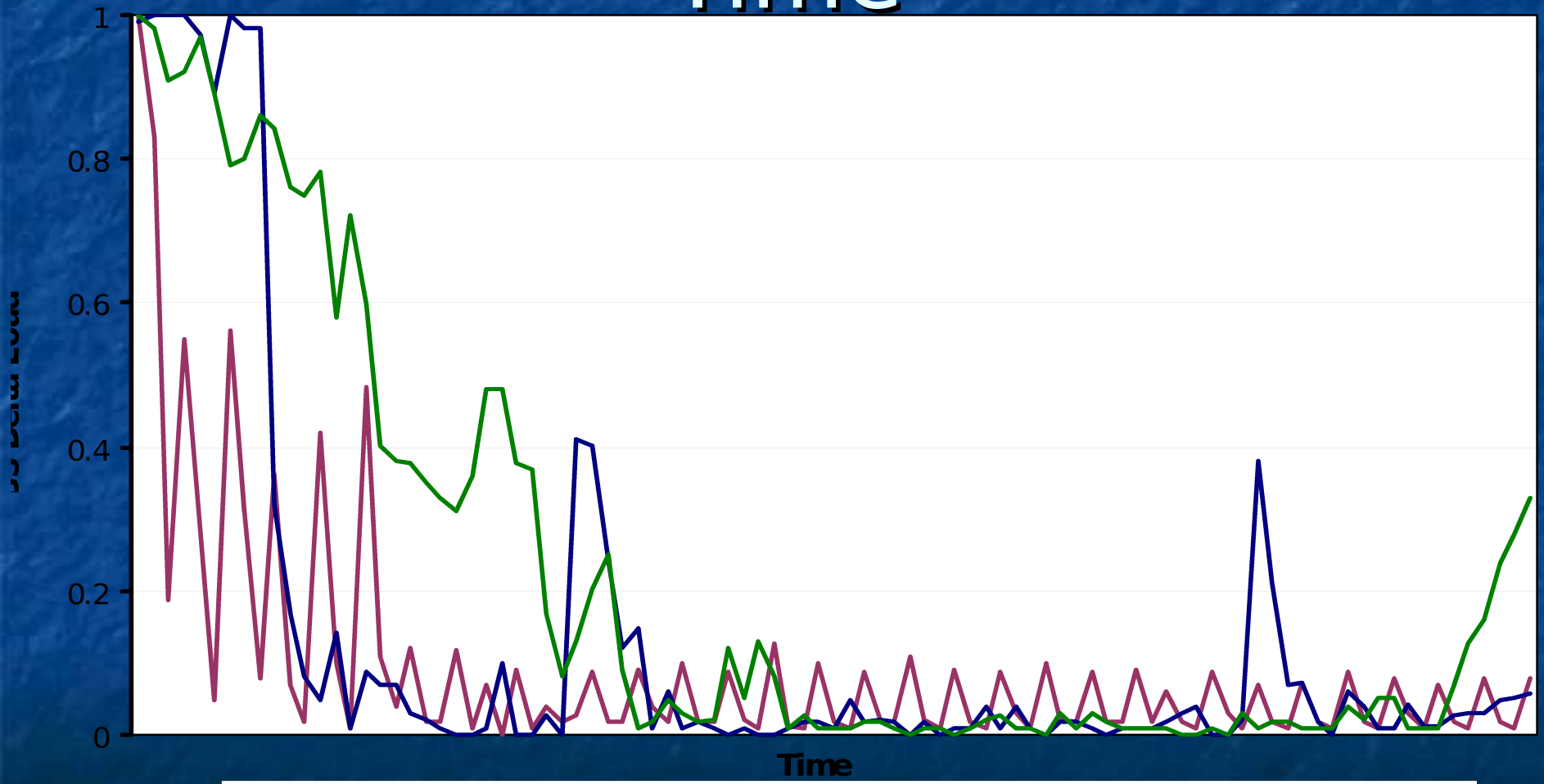
Distribution Statement A: Approved for public release; distribution is unlimited

Processors' Workload Over Time



Distribution Statement A: Approved for public release; distribution is unlimited

Processors' Workload Over Time



Distribution Statement A: Approved for public release; distribution is unlimited

Summary

- Corroborate existing studies
- Need for application level performance measurement
  - Object level is better
- At first blush, Load Balancing is a good idea
  - But…

Distribution Statement A: Approved for public release; distribution is unlimited

Acknowledgements

- This publication made possible through support provided by DoD High Performance Computing Modernization Program (HPCMP) Programming Environment & Training (PET) activities through Mississippi State University under the terms of Agreement No. # GS04T01BFC0060. The opinions expressed herein are those of the author(s) and do not necessarily reflect the views of the DoD or Mississippi State University.

Distribution Statement A: Approved for public release; distribution is unlimited